

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 8月26日

出 願 番 号

Application Number:

特願2002-244471

[ ST.10/C ]:

[ JP2002-244471 ]

出 願 人

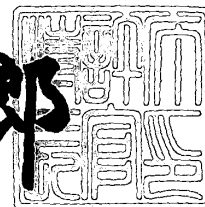
Applicant(s):

株式会社日立製作所

2003年 3月28日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3021471

【書類名】 特許願

【整理番号】 K02006671A

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/60

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 立見 彰男

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 熊▲崎▼ 裕之

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 久間 裕一郎

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】	要約書	1
【プルーフの要否】	要	

【書類名】 明細書

【発明の名称】 計算機構成変更方法およびシステム

【特許請求の範囲】

【請求項 1】

構成を変更可能な計算機システムと、該計算機のハードウェア構成に応じて課金を行うハードウェア管理システムと、該計算機のプログラム構成に応じて課金を行うプログラム管理システムとを備えた計算機構成変更システムにおいて、

上記計算機システムは、構成変更を所望する上記計算機システムのハードウェア構成とプログラム構成とを含む構成変更要求を入力した場合は、該構成変更するハードウェアの課金を変更するために上記構成変更するハードウェア構成情報を上記ハードウェア管理システムへ送り、該構成変更するプログラムの課金を変更するために上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報を上記プログラム管理システムへ送り、

上記ハードウェア管理システムは、上記構成変更するハードウェア構成情報を入力すると、該ハードウェア構成情報に基づいて該ハードウェアに対する課金を変更して上記計算機に該ハードウェアの課金情報を送り、

上記プログラム管理システムは、上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報とを入力すると、該ハードウェア構成情報と該プログラム構成に基づいて該プログラムに対する課金を変更して上記計算機に該プログラムの課金情報を送り

上記計算機は、上記ハードウェアの課金情報と上記プログラムの課金情報を入力した場合、上記構成変更するハードウェア構成情報に基づいてハードウェア構成を変更し、上記構成変更するプログラム構成情報に基づいてプログラム構成を変更することを特徴とする計算機構成変更システム。

【請求項 2】

上記計算機の構成はライセンスキーで管理されており、上記プログラム管理システムから送られた課金情報に含まれるライセンスキーに基づいて該構成を変更する手段を備えたことを特徴とする請求項 1 記載の計算機構成変更システム。

【請求項 3】

上記変更する構成変更を行う場合、上記ライセンスキーに含まれる有効期限に基づいて決定する手段を備えたことを特徴とする請求項 2 記載の計算機構成変更システム。

【請求項 4】

上記構成変更を行う場合、上記送られた課金情報とハードウェア構成およびプログラム構成との一致を判定し、一致しない場合は上記構成変更を行わない手段を備えたことを特徴とする請求項 1 記載の計算機構成変更システム。

【請求項 5】

上記構成変更を行う場合、上記構成変更するハードウェア構成に基づいて上記プログラム構成を変更する手段を備えたことを特徴とする請求項 1 記載の計算機構成変更システム。

【請求項 6】

計算機のハードウェアの構成情報を含むハードウェア構成情報と該ハードウェアのハードウェア契約更改通知先と、上記計算機のプログラムの構成情報と該プログラムのプログラム契約更改通知先とを記憶し、

構成変更を所望する上記計算機のハードウェア構成とプログラム構成とを含むハードウェア構成変更要求を受け付けた場合は、該構成変更するハードウェアの課金を変更するために上記ハードウェア契約更改通知先へ上記構成変更するハードウェア構成情報を上記ハードウェア契約更改通知先へ送り、該構成変更するプログラムの課金を変更するために上記プログラム契約更改通知先へ上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報を上記プログラム契約更改通知先へ送り、

上記プログラム契約更改通知先から送られたライセンス情報を入力した場合、上記構成変更するハードウェア構成情報に基づいてハードウェア構成を変更し、上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報に基づいてプログラム構成を変更することを特徴とする計算機構成変更方法。

【請求項 7】

上記計算機の構成はライセンスキーで管理されており、上記プログラム管理システムから送られた課金情報に含まれるライセンスキーに基づいて該構成を変更

することを特徴とする請求項 6 記載の計算機構成変更方法。

【請求項 8】

上記変更する構成変更を行う場合、上記ライセンスキーに含まれる有効期限に基づいて決定することを特徴とする請求項 7 記載の計算機構成変更方法。

【請求項 9】

上記構成変更を行う場合、上記送られた課金情報とハードウェア構成およびプログラム構成との一致を判定し、一致しない場合は上記構成変更を行わないことを特徴とする請求項 7 記載の計算機構成変更方法。

【請求項 10】

上記構成変更を行う場合、上記構成変更するハードウェア構成に基づいて上記プログラム構成を変更することを特徴とする請求項 7 記載の計算機構成変更方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

計算機のハードウェアやプログラムの構成を変更する構成変更技術に関する。

【0002】

【従来の技術】

一般的にキャパシティーオンデマンド（以下、CODと省略して記述）と呼ばれるような計算機の利用・契約の形態では、ハードウェアベンダが計算機システムのハードウェアリソース、例えばCPU、メモリ、ハードディスク（以下「HD」と略）などの一部を利用できない状態にして顧客に納入する。この時に顧客では、納入された全てのハードウェアリソースに対して料金を支払う事なく、利用可能な部分の料金を支払い使用する（正確には若干高額になる場合が多い）。顧客が計算機を長い期間利用していくと、業務処理量の増加などによりハードウェアの能力不足に直面する事がある。この時顧客は、それまで利用できなかったCPUなどのハードウェア資源を、顧客の指示により活性化し利用する事ができる。そしてハードウェアベンダに対して活性化した部分の使用料を支払う。

【0003】

CODが利用されるようになる前は、顧客がハードウェアの処理能力を増やしたいと考えた時、ハードウェアベンダによる増設工事や、それに伴う運用の停止が必要であり、時間がかかるという問題があった。一般にCODのような構成変更技術では、コマンド投入や簡単な装置の設定によりハードウェアを活性化させ、リブートを行うだけでハードウェアが増強された状態になる。実装例は少ないがリブートを必要とせずに能力の向上ができる物も知られている。

【0004】

CODではハードウェアの活性化の指示を行った時に、ハードウェアベンダに対して、ネットワークを通じて「CODによるハードウェアの活性化」が発生した事を伝える方法がある。これによりハードウェアベンダでは、計算機使用料をCOD発動後のハードウェア構成に基づいて請求する事が可能である。また、不正使用を防ぐ為に、COD形態で契約した顧客のシステムがハードウェアベンダとの通信ができない状態になった場合には、全てのハードウェア資源を活性化しているとみなして課金する方法が知られている。

【0005】

【発明が解決しようとする課題】

従来のCODでは、ハードウェアの能力増強に関しては、部品の到着や増設工事を待つ必要がない分だけ短時間で増強する事ができた。しかしそのハードウェア上で動作しているプログラムには、以下の2つの問題が発生する事があった。

【0006】

第1に、ハードウェアリソースの追加により、そのハードウェア上で動作するプログラムがハードウェアリソースの追加による能力向上分を利用できるとは限らない事である。例えばCPUの数だけプロセスを起動して、全てのCPUを同時に使うような並列アプリケーションプログラムを想定した場合、CODによりCPUが増えているにも関わらず、アプリケーションプログラムが自動的に全てのCPUを利用するようになる事はなかった。このため顧客は手動でアプリケーションプログラムの停止や再起動、構成定義の変更を行う必要があった。

【0007】

第2に、不正利用を防ぐために、上記プログラム自身がハードウェア構成を認

識し、契約した構成以上のハードウェアの上では動作しないように保護をかけているという問題がある。この保護には一般にライセンスキーと呼ばれるデータが使われる。この場合のライセンスキーは、プログラムに対して実行が許可されているハードウェアの構成を伝える役割を持ち、プログラムはライセンスキーの意味するハードウェア構成の上でのみ動作する実装になっている。例えばCPUの数によって使用料金が変わるプログラムにおいては、実行されている計算機のCPUの数を得て、ライセンスキーに含まれる契約以上のCPU数の計算機上での動作を停止するプログラムや、ライセンスキーが許す範囲までの実行能力に制限するプログラムがある。このようなプログラムを動作させているシステムで、適切なライセンスキーを入手する前に間違ってCPUを増やしてしまった場合、ライセンスキーを入手するまで業務が行えなくなる等の事態が発生する。また、顧客自身があらかじめCOD発動後のライセンスキーを入手しておかなければならないと認識していた場合も、CODを発動したいと考えた後に、各アプリケーションプログラムベンダに連絡し、料金交渉を行い、ライセンスキーが配布されてくるのを待たなければならないというタイムラグが発生してしまい、CODの本来の目的である即時能力拡張が実現できなかった。

## 【 0 0 0 8 】

つまり現在のCODは「短時間でのハードウェア増設」だけを実現しており、本質的な課題である「顧客業務に対する処理能力の短時間での増強」を実現できていない。

## 【 0 0 0 9 】

本発明の目的は、ハードウェアの構成変更を反映したプログラムの能力変更を実現する計算機構成変更方法およびシステムを提供することにある。

## 【 0 0 1 0 】

## 【課題を解決するための手段】

本発明では上記課題を改善するために、構成を変更可能な計算機システムと、該計算機のハードウェア構成に応じて課金を行うハードウェア管理システムと、該計算機のプログラム構成に応じて課金を行うプログラム管理システムとを備えた計算機構成変更システムにおいて、以下の処理を行う。



(1) 上記計算機システムは、構成変更を所望する上記計算機システムのハードウェア構成とプログラム構成とを含む構成変更要求を入力した場合は、該構成変更するハードウェアの課金を変更するために上記構成変更するハードウェア構成情報を上記ハードウェア管理システムへ送り、該構成変更するプログラムの課金を変更するために上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報を上記プログラム管理システムへ送り、

(2) 上記ハードウェア管理システムは、上記構成変更するハードウェア構成情報を入力すると、該ハードウェア構成情報に基づいて該ハードウェアに対する課金を変更して上記計算機に該ハードウェアの課金情報を送り、

(3) 上記プログラム管理システムは、上記構成変更するハードウェア構成情報と上記構成変更するプログラム構成情報とを入力すると、該ハードウェア構成情報と該プログラム構成に基づいて該プログラムに対する課金を変更して上記計算機に該プログラムの課金情報を送り

(4) 上記計算機は、上記ハードウェアの課金情報と上記プログラムの課金情報を入力した場合、上記構成変更するハードウェア構成情報に基づいてハードウェア構成を変更し、上記構成変更するプログラム構成情報に基づいてプログラム構成を変更する

以上により、ハードウェアの構成変更を反映したプログラムの能力変更を実現する計算機構成変更方法およびシステムを実現することができる。

【 0 0 1 1 】

【発明の実施の形態】

図 1 は発明の全体を示すブロック図である。CODでの利用契約を適用した顧客システム 10、HWベンダ 20、APベンダ 30、38、39、がある。本実施例では、CODを構成変更要求もしくは能力変更要求の一例として説明しているが、CODに限定されるものではなく、構成変更要求もしくは能力変更要求により、ハードウェアやプログラムの構成や能力が変更しても良い。

【 0 0 1 2 】

図 2 は図 1 で示したブロックのうち、顧客システム 10、HWベンダ 20、APベンダ 30 の通信がどのような順番で起きるかを示した図であり、以降で説明

する処理の流れの概要を示している。COD発動コマンド（構成変更要求もしくは能力変更要求でも良い）が投入されるとCOD発動指示受付部11が検出し、HW-COD発動制御部12がHWベンダ20内のCOD発動通知受付部21にCODが発動された事を通知する。次にAP-COD発動制御部14がAPベンダ30内のCOD発動通知受付部31にCODが発動された事を通知する。APベンダ30からは顧客システム内のAPライセンス管理部14にライセンスキーが送信される。しばらく後にHW課金管理部22とHW料金支払い部13の間で、料金の請求とそれに対する支払いが行われる。さらにHW料金の支払いと独立して、AP課金管理部とAP料金支払い制御部の間で、料金の請求とそれに対する支払いが行われる。

## 【0013】

図1で、顧客システムで顧客がCOD発動を指示すると、COD発動指示受付部11が検出する。COD発動指示受付部11ではHW-COD発動制御部12に対して指示を出す。

## 【0014】

図3はHW-COD発動制御部12とCOD発動通知受付部21のフローチャートである。まずHW-COD発動制御部12ではCOD発動指示受付部11より指示を受けた後、ステップ121でハードウェアの構成を変更する。次にステップ122でHWベンダ20に対して、CODの発動と、COD発動によりハードウェア構成がどのように変化したかを通知する。

## 【0015】

HWベンダ20では、ステップ122によるCOD発動の通知をCOD発動通知受付部21にて受信する。ステップ211でCODの発動情報を受け取った後、ステップ212で各顧客システムのハードウェア構成がどのようなになっているかを管理しているCOD-DB23に反映する。

## 【0016】

図8はHWベンダ20内のCOD-DB23の構造である。顧客システム識別子2301に対して、顧客毎、つまり顧客A、顧客Xおよび顧客Yに対するエントリがそれぞれエントリ2310、2320および2330がある。それぞれの

顧客に対して、現在のCODの発動状況を示すCPUの個数2302とメモリ容量2303とHD量2304と、領収済み料金2305が格納されている。また、顧客毎にCODの発動状況を示す指標であるCPU2312、メモリ2313、HD2314の値と、それぞれのCODの状態に対応した料金2315が格納される。顧客Aについては、現在の契約はCPU数2ヶ、メモリ容量8GB、HD容量100GBであるため、顧客A契約2311から1900円が使用量となる。図8に示すように課金する項目が顧客A、顧客Xおよび顧客Yで異なるようにしても良い。

## 【0017】

計算機の使用料金の請求は、このCOD-DB23に格納されているCOD発動状況毎の料金に基づいて課金管理部22とHW料金支払い部13の間で行われる。

## 【0018】

CODによるハードウェアリソースの増減は、一旦リブートをしなければ有効にならない実装をされている事が多いが、リブートを行わずに、リソース量を変更することも可能である。

## 【0019】

本発明の特徴は、CODの発動通知をAPベンダ30、38、39にも行う事である。顧客システム10にはCODを発動した場合に発動を通知すべきアプリケーションプログラムのベンダを全て含む通知先一覧15を保持しており、そこに登録されている全てのベンダに通知する。図1では例としてAPベンダを3つ書いているが、38、39については内容が30と同一の為、記述を省略している。以降ではAPベンダ30についてのみ説明を行うが、他のベンダ38、39についても同一の内容である。ここでアプリケーションプログラムを一例に説明をするが、本発明は、プログラム一般に適用可能であり、アプリケーションプログラムに限定されるものではない。

## 【0020】

COD発動指示受付部11では、CODの発動を検出すると、AP-COD発動制御部14にも指示を行う。図4は顧客システム内のAP-COD発動制御部

14と、APベンダ内のCOD発動通知受付部31と、顧客システム内のAPライセンス管理部16のフローチャートである。AP-COD発動制御部14ではステップ141でCOD発動時の通知先一覧15から通知先を得て、ステップ142でそれぞれの通知先に、COD発動後のハードウェア構成がどのように変化しているかを通知する。

#### 【0021】

APベンダ30では、AP-COD発動通知受付部31で通知を受け付ける。ステップ311でCODが発動された事と発動後のハードウェア構成を受信してから、ステップ312で各顧客のハードウェア構成がどのようなになっているかを管理しているCOD-DB33に反映する。その後ステップ313でCODの発動状況から対応するライセンスキー17を生成して、ステップ314で顧客システム10のAPライセンス管理部16に対して送信する。ライセンスキー17は、CPU、メモリ、HDなどCODの対象となるハードウェアリソースの状態と有効期限などから構成される。これらは、暗号化される実装も考えられる。また、あらかじめ乱数等で定義してCOD-DB33内に格納しておく方法も考えられる。

#### 【0022】

図9はAPベンダ内のCOD-DB33の構造である。顧客システム識別子3301に対して顧客毎のエントリ3310、3320、3330がある。それぞれの顧客に対して、現在のCODの発動状況を示すCPUの個数3302とメモリ量3303とHD量3304、領収済み料金3305と、ライセンスキーの残り有効期間3306が格納されている。また、顧客毎に、COD発動状況を示す指標であるCPU3312、メモリ3313、HD3314の値と、そのCOD状況に対応した料金3315が格納されている。ライセンスキーの残り有効期間3306が格納されている。顧客Aについては、現在の契約はCPU数2ヶ、メモリ容量8GB、HD容量100GBであるため、顧客A契約3311から1900円が使用量となる。図9に示すように課金する項目が顧客A、顧客Xおよび顧客Yで異なるようにしても良い。

#### 【0023】

顧客システム 1 0 では、A P ライセンス管理部 1 6 でライセンスキー 1 7 を受信する。ステップ 1 6 1 でライセンスキーを受け取った後、ステップ 1 6 2 でライセンスキー 1 7 を格納する。さらにステップ 1 6 3 でアプリケーションプログラム 4 0 に対してライセンスキーが変更になった事を通知する。ライセンスキー 1 7 は通常、暗号化されており構成定義ファイルの中に格納される実装が行われる事が多いが、本発明ではその方法については規定しない。

## 【 0 0 2 4 】

本発明では上記のように、A P ライセンス制御部 1 6 のステップ 1 6 3 において、アプリケーションプログラム 4 0 に対して C O D の発動を通知する事も特徴としている。

## 【 0 0 2 5 】

さらに、C O D を発動した後、アプリケーションプログラム 4 0 や O S を停止せずにハードウェアリソースを増減できる場合、かつ、アプリケーションプログラム 4 0 自体も動作中に使用するハードウェアリソース量を増減できる場合に、アプリケーションプログラム 4 0 を停止する事なく、C O D を発動した後のハードウェアリソースを使う事ができるようにする事も特徴とする。この実施例は後述する。

## 【 0 0 2 6 】

次にアプリケーションプログラム 4 0 の実施例として図 6、図 7 の 2 つの形態を示す。以下で述べるアプリケーションプログラムは実施の一形態であり、本発明の範囲を限定する物ではない。

## 【 0 0 2 7 】

図 6 はアプリケーションプログラム 4 0 の実施例の一形態を示すフローチャートである。アプリケーションプログラム 4 0 は起動時にステップ 4 0 1 で H W 構成取得部 1 9 からハードウェアの構成を得て、ステップ 4 0 2 でライセンスキー 1 7 を得る。次にステップ 4 0 3 で、そのライセンスキー 1 7 が意味しているアプリケーションプログラム 4 0 の動作が許可されたハードウェアの構成と、動作しようとしている実際のハードウェアの構成とを比較して、不当な利用ではないかのチェックを行う。例えばライセンスキー 1 7 が 4 C P U までの実行を許可し

ていて、実際のCPUが8個であった場合、不当利用であると判定される。この時はステップ406で例えばアプリケーションプログラム40を停止する。逆に、例えばハードウェアの実際の構成とライセンスキー17に示されている構成が一致した場合は正当利用だと判定され、ステップ404でライセンスキー17内に含まれている有効期間のチェックを行う。ライセンスキー17の有効期限が切れていれば不正利用であるので、ステップ406でアプリケーションプログラム40を停止する。有効期間内であれば、ステップ405にてアプリケーションプログラム40の本来の作業である業務処理を行う。なおライセンスキー17の有効期限は時間の経過と共にチェックしなおさねばならないため、ステップ405の業務処理は例えば一日一回などの頻度で一時中断し、ステップ404に戻り有効期限チェックを行う。

## 【0028】

次に、図7はアプリケーションプログラム40の実施例の図6とは異なる一形態を示すフローチャートである。この例では前述した「CODの発動に応じて稼動中にハードウェアの使用量を変更できるアプリケーションプログラム」を示している。

## 【0029】

ステップ401、ステップ402は図6と同様、ハードウェア構成とライセンスキー17を得る処理である。この実施例のアプリケーションプログラム40はライセンスキー17が意味しているハードウェアの構成情報を用いて最適なハードウェア使用量を決める。この方法として一実施例をステップ403、407、408で示している。ステップ403で、ライセンスキー17が意味しているアプリケーションプログラム40の動作を許可したハードウェアの構成と、動作しようとしている実際のハードウェア構成を比較する。もしライセンスキー17に示されているハードウェア構成以上のハードウェア上で起動されるという状態であれば、ステップ407にてアプリケーション40が利用するハードウェア量がライセンスキー17に従って決定される。逆にライセンスキー17のほうが実際のハードウェア構成より大きければ、ステップ408にて、アプリケーション40が利用するハードウェア量が実際のハードウェア構成を使い切るように決定さ

れる。例えば、ライセンスキー17が4CPUまでの動作許可を意味していた場合、ハードウェア構成が8CPUであれば、プロセスを4個だけ起動する。逆にライセンスキー17が8CPUまでを意味しており、ハードウェア構成が4CPUであった場合は、プロセスを4個だけ起動する。そしてステップ405の業務処理を行う。本実施例でいう構成変更は、容量変更や能力変更も含んでいる。

## 【0030】

CODの発動にともない、ステップ405の業務処理中に、ライセンスキー17が変更された事を通知する割り込み409が発生する。この場合は割り込み処理によって一旦ステップ401の前に処理に戻る。

## 【0031】

以下に図7の各ステップにそって、メモリを増加させた場合と、CPUを増加させた場合の例を二つ示す。

## 【0032】

メモリを増加させる例として、実装メモリに応じて業務処理の作業領域を決定するタイプのアプリケーションプログラム40を想定する。このアプリケーションプログラム40では、OSの使用量やアプリケーションプログラム40自体のロードモジュールや固定領域の為に、0.5GB+実装メモリの1/4を利用不能領域として引き、残りの部分を業務処理に割り当てるように実装されているとする。搭載メモリ4GBの計算機の場合、ステップ401でメモリ量4GBという情報が得られ、ステップ402でライセンスキー17が4GBメモリでの実行を意味している事を得られたとする。ステップ403の判定で「HW構成=ライセンスキー」であるのでステップ407に進む。ステップ407ではライセンスキー17が示す4GBに従い、利用不能領域として0.5+1GBを引き、2.5GBをアプリケーションプログラム40の作業領域として割り当て、ステップ405で2.5GBを利用して業務処理を行う。

## 【0033】

ステップ405の業務処理中にメモリを16GBに増加させるCOD発動を行うと、ハードウェアの構成が16GBになりライセンスキー17が16GBを意味する状態になった後に、ステップ409の割り込みがアプリケーションプログ

ラム40に対して発生する。すると順にステップをたどり、ステップ407によりアプリケーションプログラム40は16GBで動作する時の設定である、利用不能領域として0.5+4GBを引いた残り11.5GBをアプリケーションプログラム40自体の作業領域として割り当てなおす。結果としてステップ405の業務処理は作業領域が2.5GBから11.5GBへ増えたので、アプリケーションプログラム40は業務の処理能力を向上させる事ができる。

## 【0034】

なお、一般的にアプリケーションプログラムでは仮想メモリ空間しか確保できないため、仮想メモリの大きさと実メモリの使用量が1対1になっているとは限らない。しかし、実メモリからOSの使用量など固定的に必要な容量を引いたメモリ量までの仮想メモリを使用すると、システム上で利用されている全ての仮想メモリ量を実メモリ以下におさえる事ができる。この状態ではアプリケーションプログラムが利用する領域は全て実メモリ上に収まり（ページアウトされる事がなくなり）、高効率で動作することができる。

## 【0035】

CPUを増加させる例として、実装されているCPUの数と同じ数のプロセス（処理単位の例でありスレッドであるケースなどもある）を起動して、業務業務処理を行うタイプのアプリケーションプログラム40を想定する。搭載CPU4個の計算機の場合、ステップ401でCPU数4個という情報が得られ、ステップ402でライセンスキー17がCPU4個での実行を意味している事を得られたとする。ステップ403の判定で「HW構成＝ライセンスキー」であるのでステップ407に進む。ステップ407ではライセンスキー17が示す4CPUに従い、プロセスを4個起動し、ステップ405で4個のプロセスを利用して業務処理を行う。

## 【0036】

ステップ405の業務処理中にCOD発動が行われCPUを8個に増加すると、ハードウェアの構成が8CPUになりライセンスキー17が8CPUを意味する状態になった後に、ステップ409の割り込みがアプリケーションプログラム40に対して発生する。すると順にステップをたどり、ステップ407によりア



アプリケーションプログラム40は8CPUで動作する時の設定である、プロセス8個を起動しなおす。結果としてステップ405の業務処理を処理するプロセスが4個から8個に増えたので、アプリケーションプログラムは業務の処理能力を向上させる事ができる。

## 【0037】

なお、図7と図6は分けて記述しているが構成要素を簡潔に説明するためであり、一つの実施形態に上記2つの方式を用いて適用することができる。例えば図6中のステップ404、406の処理（ライセンス有効期限チェック）は図7に適用可能であるし、ステップ403、406の処理（不正使用時のアプリケーション停止）も図7に適用可能である。

## 【0038】

COD発動後、APベンダ30は各顧客に対して新しいハードウェア構成に対応した料金の支払いを求める。図1におけるAP課金管理部32はAPベンダ内に書かれているが、実際には再販業者などAPベンダ30とは別の企業またはシステムの構成を採ることも可能である。AP課金管理部32は顧客10のAP料金支払い制御部18に対して、新しいハードウェア構成に対応した料金の支払いを要求する。

## 【0039】

図5はAP課金管理部32のフローチャートである。初めにステップ321でCOD-DB33を検索し、顧客にどれだけの金額を請求するかをCOD状態毎の料金3315から検出し、ステップ322にて料金を請求する。次に例えば毎日12時などの契機で料金の支払いが行われたかをステップ323でチェックする。支払いが無い場合はステップ328でCOD-DB33からライセンスキーの有効期間3306を調べ、有効期限が過ぎていれば、ステップ329にて利用権限が全く無い事を意味するライセンスキーを送り込む。有効期限が切れていなければステップ323に戻り、再び支払いを待つ。なお、支払いがあった場合はステップ324にてCOD-DB33内の領収済み料金3305を更新する。さらにステップ325にて、COD状態毎の料金3315と比較して支払われた料金に不足があるかを調べる。不足があった場合はステップ323の支払い待ちに

戻る。不足がなかった場合には次回の支払い期限を決め、ステップ 3 2 6 にて顧客に対して新しい有効期間に対応するライセンスキーを送り、ステップ 3 2 7 にて COD-DB 3 3 内の有効期間 3 3 0 6 を更新する。

【 0 0 4 0 】

有効期限の情報が含まれていないライセンスキーで利用を制御していた場合、COD 発動後のハードウェア構成に対応したライセンスキーを入手された直後に、通信できないように通信路が切断された場合は、例え不正使用が行われていても AP ベンダ 3 0 からアプリケーションプログラム 4 0 の実行制限を行う事ができないという問題が発生する。それに対して、ライセンスキー 1 7 自体に有効期限を持たせる事で不正利用を防ぐ事ができる。ライセンスキー 1 7 の有効期限が切れていた場合には、アプリケーションプログラム 4 0 はステップ 4 0 4 の判定により動作しないようになり、COD 発動後の状態での不正利用を防ぐ事ができる。

【 0 0 4 1 】

この場合、ライセンスキーの自動更新を行う実施例を以下に示す。ひとつは、ライセンスキー 1 7 の期限切れを検出したアプリケーションプログラム 4 0 自身が、AP ベンダ 3 0 に新しいライセンスキーを要求して、入手する実装がある。また、AP ベンダ 3 0 が、自身の持つ COD-DB 3 3 内のライセンスキー有効期限 3 3 0 6 を参照し、ライセンスキー 1 7 を送り込む方法もある。いずれの場合でも通信路の切断ではライセンスキー 1 7 の更新ができなくなるため、顧客システム 1 0 内にあるライセンスキー 1 7 の有効期限が切れるため、不正利用を防ぐ事ができる。本発明では上記ライセンスキー更新処理の実施の形態は規定しない。

【 0 0 4 2 】

なお、COD 対応のシステムを導入し本番稼動に入る前に、数ヶ月程度一時的に COD を発動して業務アプリケーションプログラムを含めたシステム全体の動作検証や、COD を発動した後の状態での性能検証を行う事が一般的であると想定される。この場合も本発明で述べてきた手段により、一時的な COD 発動用のライセンスキーの送受信を行う事ができる。

【 0 0 4 3 】

以上により、COD形態で導入したハードウェアの上で動作する業務処理アプリケーションプログラムは、CODが発動された後すぐに、CODによって増強されたハードウェアリソースを利用することができ、顧客業務に対する処理能力の短時間での増強が達成できる。また、ハードウェアリソースの増加を行った後に、アプリケーションプログラムに対して適切な使用料を支払わず、不当な利用を続けた場合に、適切な支払いが行われるまでアプリケーションプログラムの処理能力を制限や停止する手段をあたえる事で、不正利用に対する保護が可能になる。

【 0 0 4 4 】

【発明の効果】

本発明により、ハードウェアの構成変更を反映したプログラムの能力変更を実現することが可能となる。

【図面の簡単な説明】

【図 1】 本発明の実施例の全体の構成を示すブロック図である。

【図 2】 通信を行う各ブロックがどのような順番で動作するかを示したタイミングチャートである。

【図 3】 顧客システム内のHW-COD発動制御部とHWベンダ内のCOD発動通知受付部のフローチャートである。

【図 4】 顧客システム内のAP-COD発動制御部とAPベンダ内のCOD発動通知受付部と、顧客システム内のAPライセンス管理部のフローチャートである。

【図 5】 APベンダ内のAP課金管理部のフローチャートである。

【図 6】 APの実施例の一形態である。

【図 7】 図 6 とは異なるAPの実施例の一形態である。

【図 8】 HWベンダ内のCODに関する情報を保持するCOD-DBの図である。

【図 9】 APベンダ内のCODに関する情報を保持するCOD-DBの図である。

【符号の説明】

- 1 0 : 顧客システム
- 1 1 : COD発動指示受付部
- 1 2 : HW-COD発動制御部
- 1 3 : HW料金支払い制御部
- 1 4 : AP-COD発動制御部
- 1 5 : COD発動時の通知先一覧
- 1 6 : APライセンス管理部
- 1 7 : ライセンスキー
- 1 8 : AP料金支払い制御部
- 1 9 : HW構成取得部
- 2 0 : HWベンダ
- 2 1 : HWベンダ内、COD発動通知受付部
- 2 2 : HW課金管理部
- 2 3 : HWベンダ内、COD-DB
- 3 0 : APベンダその1
- 3 1 : APベンダ内、COD発動通知受付部
- 3 2 : AP課金管理部
- 3 3 : APベンダ内、COD-DB
- 3 8 : APベンダその2
- 3 9 : APベンダその3
- 4 0 : アプリケーションプログラム
- 2 3 1 0、2 3 2 0、2 3 3 0 : HWベンダ内COD-DBにある顧客の現在の  
状態を示すレコード
- 2 3 0 1 : HWベンダ内COD-DB内の顧客システム識別子欄
- 2 3 0 2、2 3 0 3、2 3 0 4 : HWベンダ内COD-DB内の現在のCOD発  
動状況欄
- 2 3 0 5 : HWベンダ内COD-DB内の領収済み料金欄
- 2 3 1 2、2 3 1 3、2 3 1 4 : HWベンダ内COD-DB内の顧客毎のCOD

発動状況の指標

2315 : HWベンダ内COD-DB内の顧客毎のCOD発動状況毎の料金一覧

3310、3320、3330 : APベンダ内COD-DB内の顧客毎の現在の  
状態を示すレコード

3301 : APベンダ内COD-DB内の顧客システム識別子欄

3302、3303、3304 : APベンダ内COD-DB内の現在のCOD発  
動状況欄

3305 : APベンダ内COD-DB内の領収済み料金欄

3306 : APベンダ内COD-DB内のライセンスキーの残り有効期間欄

3312、3313、3314 : APベンダ内COD-DB内の顧客毎のCOD

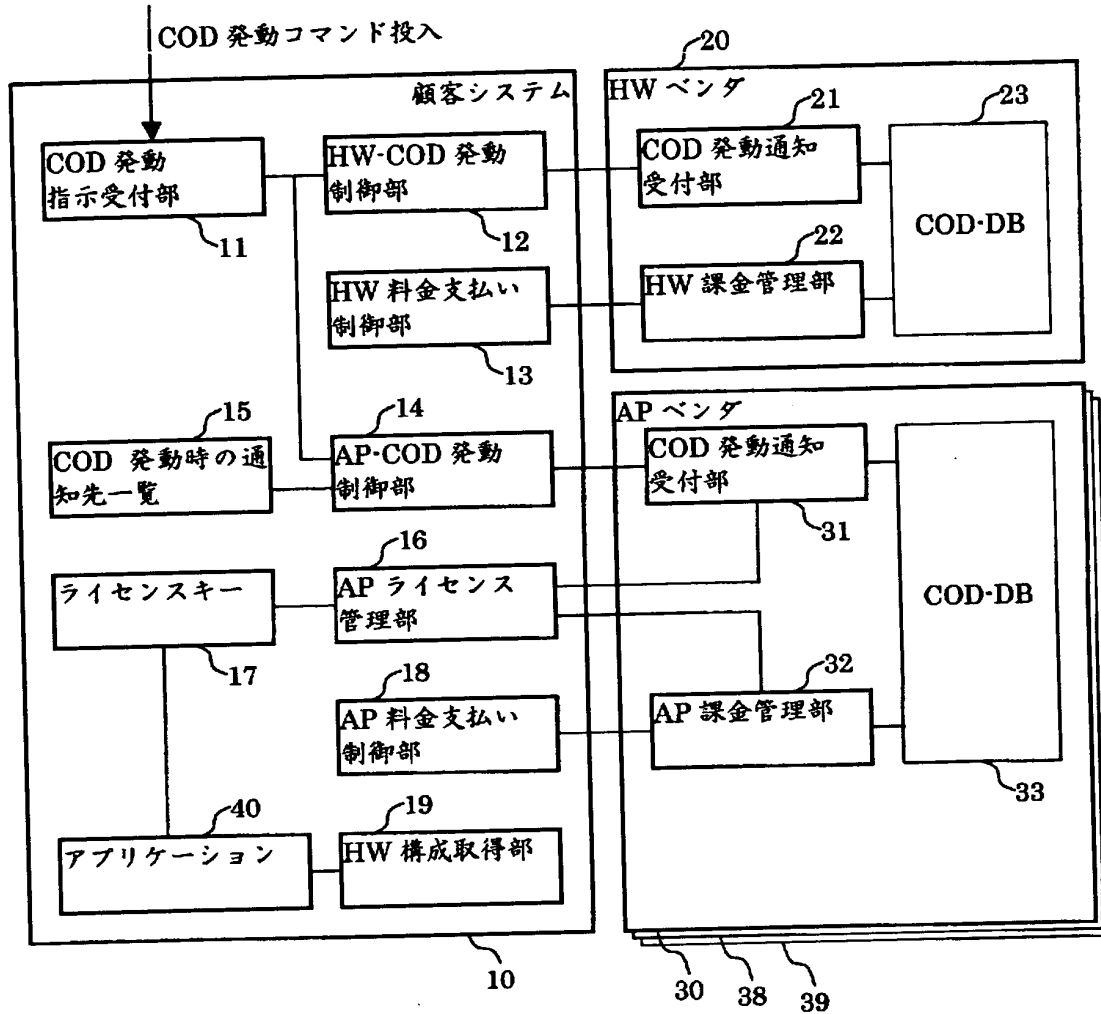
発動状況の指標

3315 : APベンダ内COD-DB内の顧客毎のCOD発動状況毎の料金一覧

【書類名】 図面

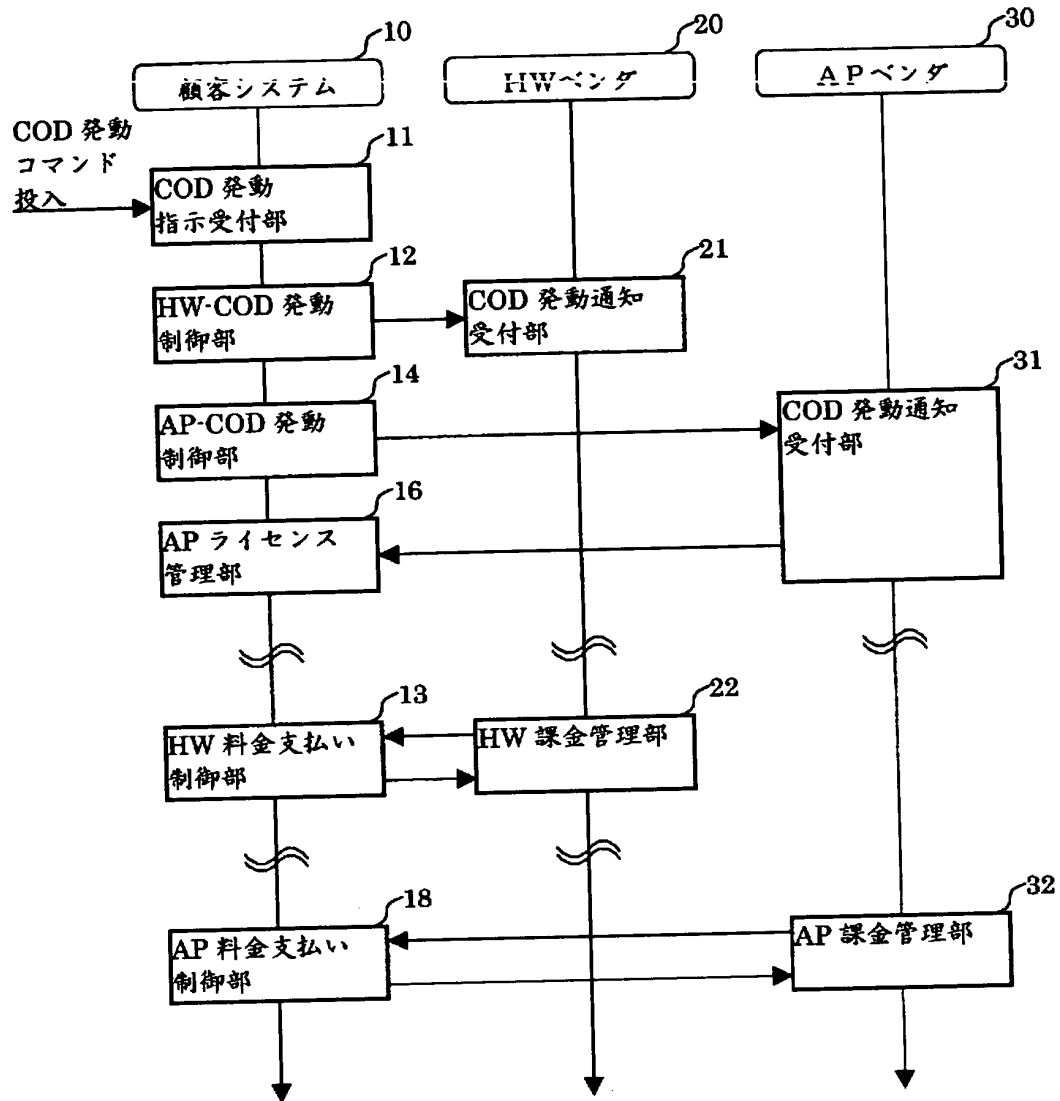
【図 1】

図 1



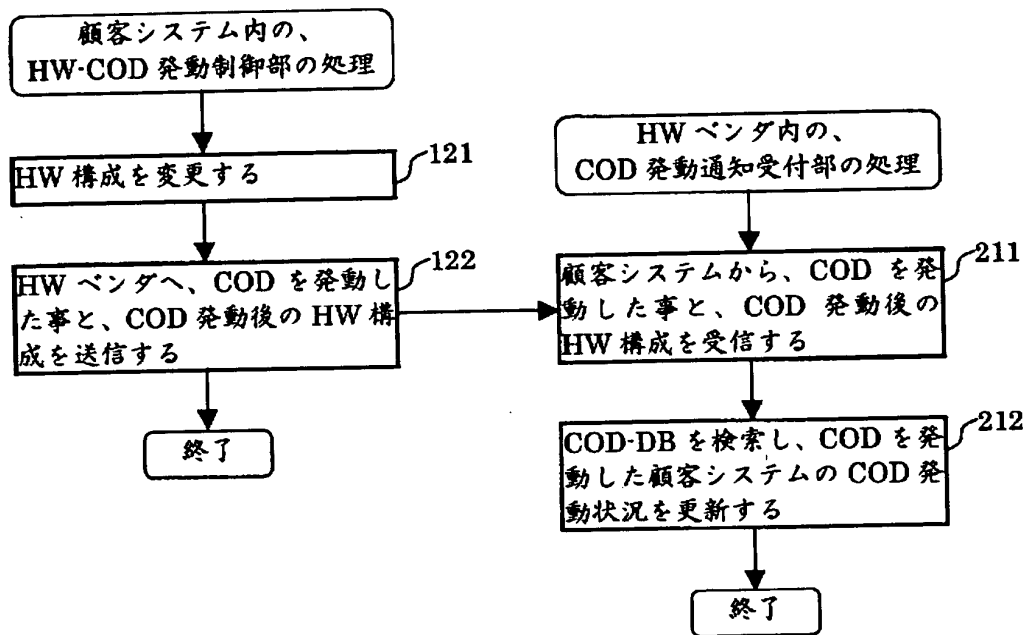
【図 2】

図 2



【図 3】

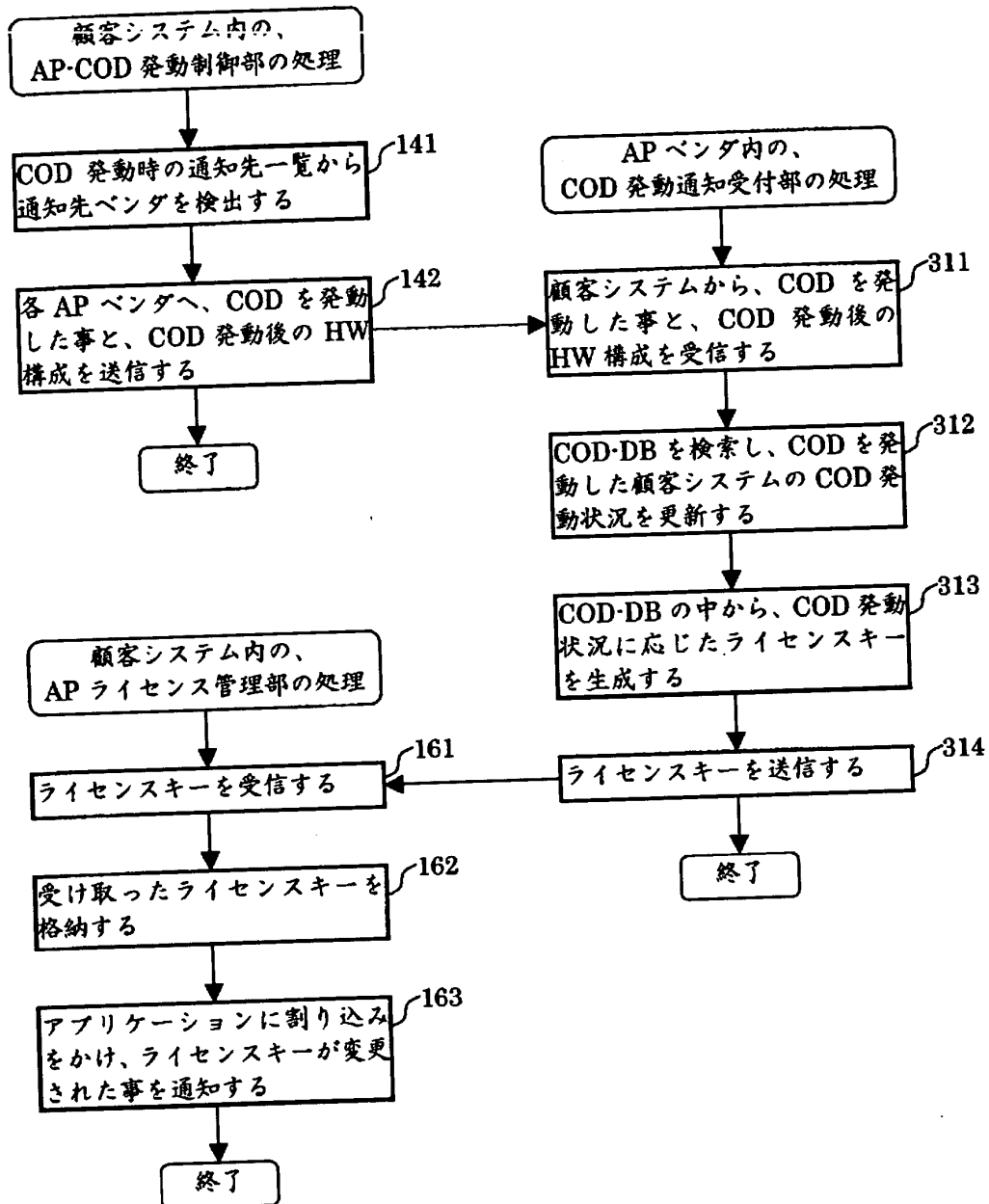
図 3





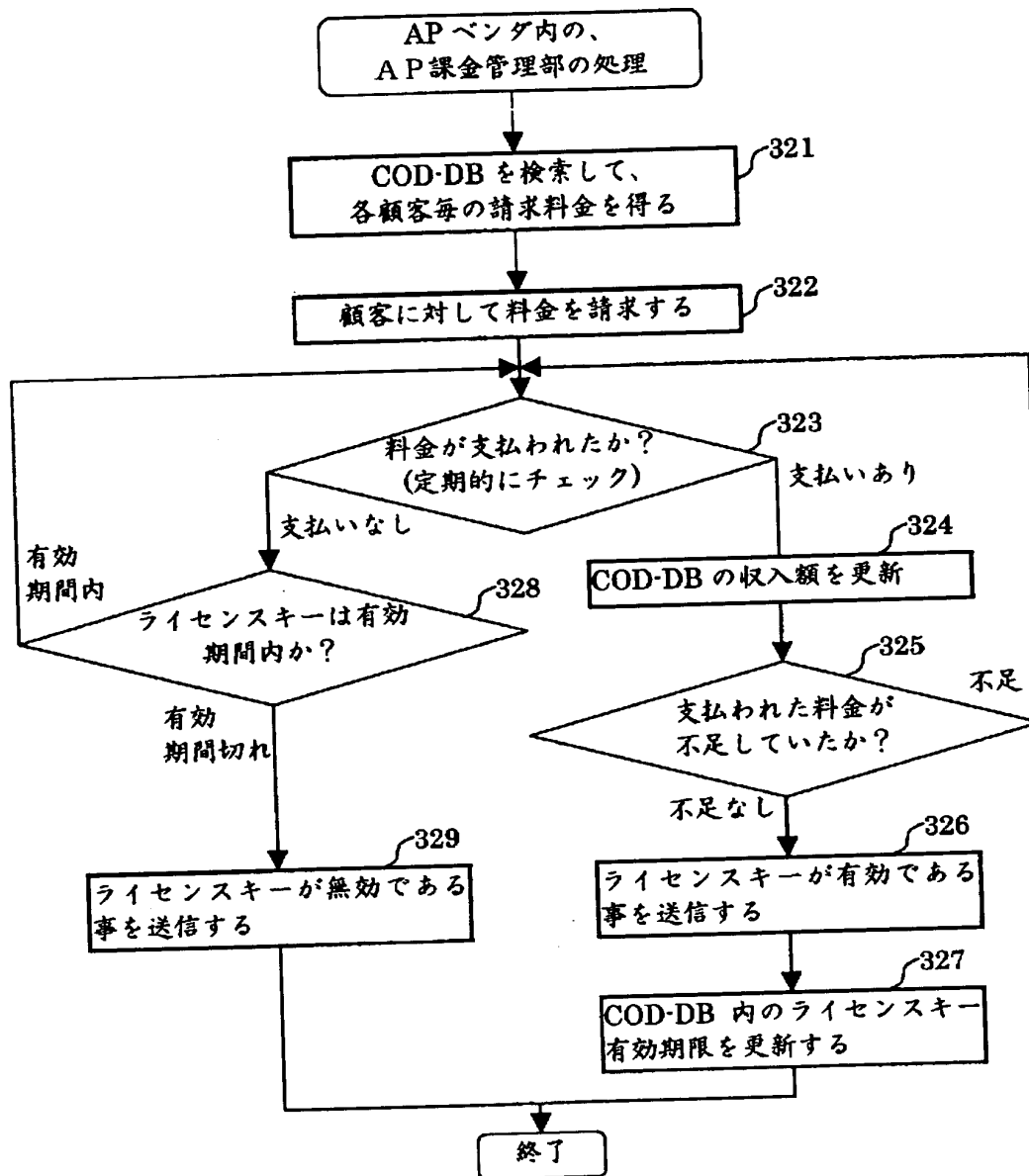
【図 4】

図 4

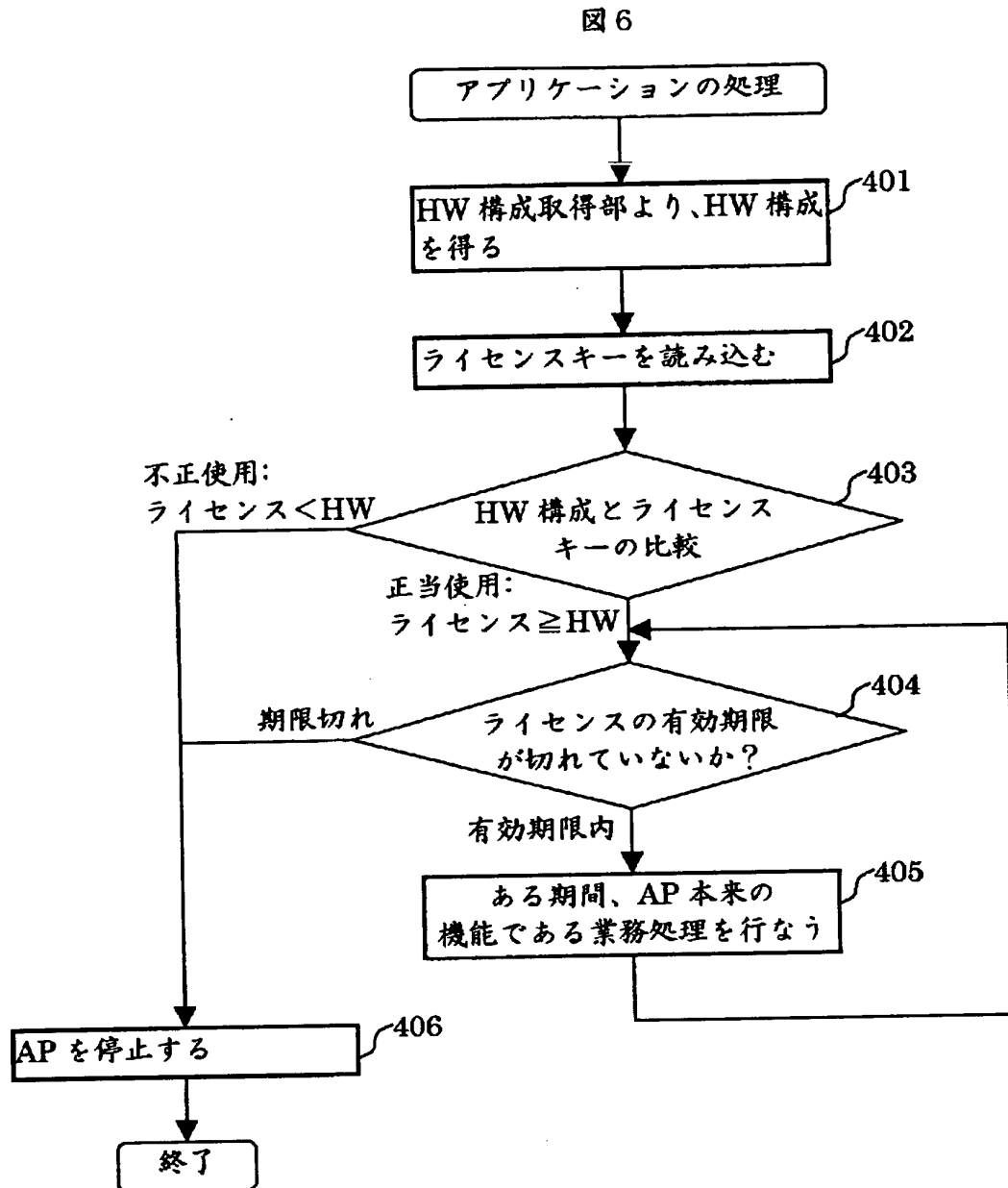


【図 5】

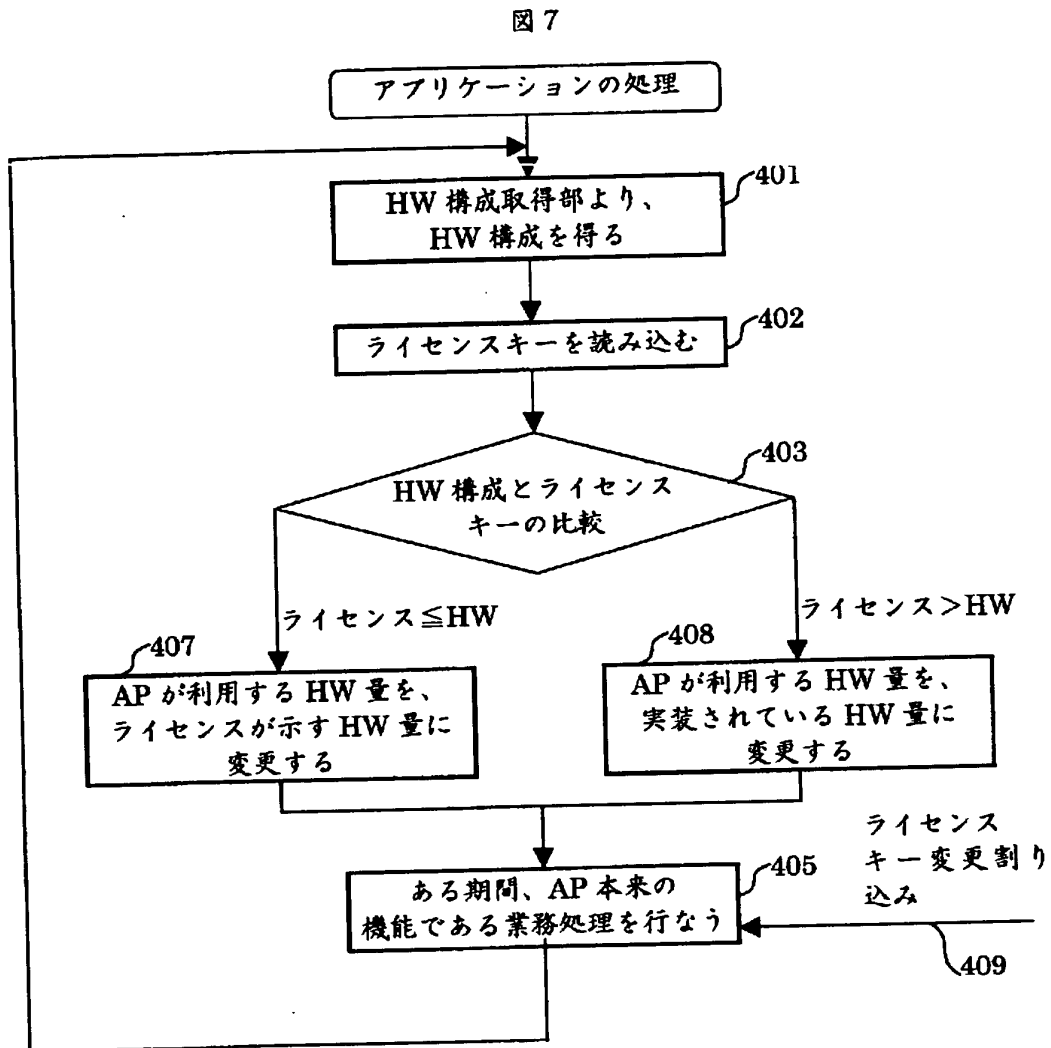
図 5



【図 6】



【図 7】



【図 8】

図 8

23				
2310		2320		2330
顧客システム識別子	顧客 A	顧客 X	顧客 Y	...
現在の cpu 数	2 個	6 個	4 個	...
現在のメモリ容量	8 GB	16 GB	変更不可	...
現在のHD容量	100 GB	変更不可	変更不可	...
領収済み料金	1800 円	0 円	2100 円	...

顧客 A の契約							
CPU	メモリ	4 GB		8 GB		16 GB	
	HD	100GB	200GB	100GB	200GB	100GB	200GB
	1 個	1000 円	1100 円	1400 円	1500 円	2200 円	2300 円
	2 個	1500 円	1600 円	1800 円	2000 円	2700 円	2800 円
	3 個	2000 円	2100 円	2400 円	2500 円	3200 円	3300 円
	4 個	2500 円	2600 円	2900 円	3000 円	3700 円	3800 円

顧客 X の契約			
CPU	メモリ	16 GB	3 GB
	2 個	...円	...円
	4 個	...円	...円

顧客 Y の契約		
CPU	2 個	...円
	4 個	...円
	6 個	...円

【図 9】

図 9

33				
3310		3320		3330
顧客システム識別子	顧客 A	顧客 B	顧客 C	...
現在の cpu 数	2 個	6 個	4 個	...
現在のメモリ容量	8 GB	16 GB	32 GB	...
現在のHD容量	100GB	100GB	変更不可	...
領収済み料金	1900 円	0 円	2100 円	...
ライセンスキーの 残り有効期間	30 日	4 日	28 日	...

顧客 A の契約							
	メモリ	4 GB		8 GB		16 GB	
	HD	100GB	200GB	100GB	200GB	100GB	200GB
CPU	1 個	1000 円	1100 円	1400 円	1500 円	2200 円	2300 円
	2 個	1500 円	1600 円	1900 円	2000 円	2700 円	2800 円
	3 個	2000 円	2100 円	2400 円	2500 円	3200 円	3300 円
	4 個	2500 円	2600 円	2900 円	3000 円	3700 円	3800 円

顧客 B の契約							
	メモリ	16 GB		24 GB		32 GB	
	HD	100GB	200GB	100GB	200GB	100GB	200GB
CPU	2 個	...円	...円	...円	...円	...円	...円
	4 個	...円	...円	...円	...円	...円	...円
	6 個	...円	...円	...円	...円	...円	...円
	8 個	...円	...円	...円	...円	...円	...円

顧客 C の契約			
CPU	メモリ	16 GB	32 GB
	2 個	...円	...円
	4 個	...円	...

【書類名】 要約書

【要約】

【課題】

従来は、アプリケーションプログラムの能力向上には、人手を介したベンダとの交渉や、ライセンスキーの入手などが必要であったため、業務処理の観点からは短時間に能力を向上する事ができなかった。

【解決手段】

CODの発動を契機として、ソフトウェアベンダに対してもCODの発動を通知する。CODの発動通知を受けたソフトウェアベンダは顧客の計算機に対して自動的に新しい構成で動作するためのライセンスキーを送信する事により、短時間での業務処理能力の向上を実現する。また、アプリケーション使用料の支払いが滞った場合には、アプリケーションベンダから制限付きのライセンスキーを送り込み、不正使用を防ぐ事ができる。さらに顧客システムではライセンスキーを受け取った契機で、アプリケーション自体に通知を行う事で、アプリケーション動作中に動的に処理能力を向上させる手段を提供する。

【選択図】 図 1

特 2 0 0 2 - 2 4 4 4 7 1

認定・付加情報

特許出願の番号	特願 2 0 0 2 - 2 4 4 4 7 1	
受付番号	5 0 2 0 1 2 5 5 3 4 4	
書類名	特許願	
担当官	第七担当上席	0 0 9 6
作成日	平成 1 4 年 8 月 2 7 日	

< 認定情報・付加情報 >  
【提出日】

平成 14 年 8 月 26 日

次頁無



出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日  
[変更理由] 新規登録  
住 所 東京都千代田区神田駿河台4丁目6番地  
氏 名 株式会社日立製作所